[Notes]
C Compiler Package for R32C Series,
C Compiler Package for M32C Series,
and C/C++ Compiler Package for M16C Series and R8C Families

R20TS0699EJ0100
Rev.1.00
Jun. 01, 2021

Overview

When using any of the products in the title, note the following point.

1. Initializing an array with a wide string literal or simple string literal

1. Initializing an array with a wide string literal or simple string literal

1.1 Applicable Products

C compiler package for R32C series V.1.01 Release 00 to V.1.02 Release 01

C compiler package for M32C series V.5.00 Release 1 to V.5.42 Release 00

C/C++ compiler package for M16C series and R8C families V.5.00 Release 1 to V.6.00 Release 00

1.2 Description

When initializing an array with a wide string literal or simple string literal, an undefined value may be stored after the terminating NULL character. The undefined value may be stored in the following addresses:

- For a wide string literal:

  An undefined value is stored in the two-byte addresses immediately after NULL.
- For a simple string literal:

  An undefined value is stored in the following addresses:
  - If the size of the simple string literal is even:

    Two bytes immediately after NULL
  - If the size of the simple string literal is odd:

    Two bytes of the NULL address +2 and +3

1.3 Conditions

This problem may arise if all of the conditions from (1) to (3) are met.

(1) An array has a two-byte element type (Note 1).

(2) A wide string literal or simple string literal is specified as a initial value for the array in condition (1).

(3) The relationship between the size (in bytes) of the array (1) and the size (in bytes) of the string literal (2) is one of the following (Note 2).

(3-1) When (2) is a wide string literal:

The size of the array (1) > The size of the wide string literal (2)

(3-2) When (2) is a simple string literal and the size is even:

The size of the array (1) > The size of the simple string literal (2)

(3-3) When (2) is a simple string literal and the size is odd:

The size of the array (1) > The size of the simple string literal (2) + 1 byte

(Note 1): In the case of C compiler package for R32C series, this includes an int type when the compilation option -fint_16 is specified.

(Note 2): The size of the string literal includes the NULL character.

1.4 Example

Below is an example of the error.

```
/* tp.c */
const unsigned short array[6] = L"abc";
```

array[4] might be an undefined value.

1.5 Workarounds

You can avoid this problem by one of the following methods:

(1) Fill the end of the initial value with '¥0' to make the initial value size equal to the array size.

Example: `const unsigned short array[6] = L"abc¥0¥0";`

(2) Adjust the array size to match the size of initial value.

Example: `const unsigned short array[4] = L"abc";`

(3) Avoid specifying the array size.

Example: `const unsigned short array[] = L"abc";`

(4) Change the array type to a type that is not covered by condition (1).

(5) Separate the array declaration and value assignment.

1.6 Permanent Measure

We do not plan to fix it.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.01.21	-	First edition issued

Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included.

The URLs in the Tool News also may be subject to change or become invalid without prior notice.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2021 Renesas Electronics Corporation. All rights reserved.

TS Colophon 4.2